# I. Backtracking folosind permutări, aranjamente, combinări

# O grupă de studenţi trebuie să-şi planifice m examene în n zile, n>=m.. In câte moduri şi cum se pot programa examenele, ştiind că într-o zi se poate programa cel mult un examen?

1. Câte numere naturale diferite se pot forma cu cifrele 1, 2, 3, …, n, dacă în fiecare astfel de număr, orice cifră apare cel mult odată şi care sunt acestea?
2. Să se afişeze tote numerele de n cifre a căror prefixe sunt divizibile cu 2.
3. Se citeşte de la tastatură un număr natural n<=14, care reprezintă dimensiunea unei table de şah. Să se genereze toate modalităţile de aranjare pe tablă a n ture, astfei încât oricare două ture să nu se atace. Rezultatele se vor afişa într-un fişier ture.txt ca în exemplu de mai jos, pentru n=2:



1. Să se afişeze toate posibilitătile de aranjare pe o casetă a n melodii, codificate cu numere naturale de la 1 la n, astfel încât melodia x să se cânte după melodia y, n maximum 10.
2. Într-o mare închisă sunt n porturi. Să se numere şi stabilească toate voiajele prin p porturi (p<=n). Porturile se introduc prin denumiri, după ce s-au citit valorile pentru n şi p. Soluţiile vor fi afişate secvenţial, pentru a putea fi citite (apăsare tastă) , numerotat.
3. Se citesc de la tastatură un număr natural n(n<=20) şi un număr natural v (v<n). Să se scrie un program care afişează toate numerele de la 1 la n în toate modurile posibile astfel încât între oricare două numere succesive diferenţa în modul să fie mai mare decât v. Datele de ieşire se vor scrie în fişierul ieşire.txt.
4. Se citeşte un număr natural n. Să se genereze toate matricile pătratice binare de dimensiune nxn, cu proprietatea că în fiecare coloană există exact o cifră 0.

Ex. Pentru n=3

0 0 0 0 0 1 0 0 1

1 1 1 1 1 0 1 1 1

1 1 1 1 1 1 1 1 0

1. Un grup de n (n<=10) persoane numerotate de la 1 la n sunt aşezate pe un rând de scaune, dar între oricare două persoane vecine s-au ivit conflicte. Scrieţi un program care afişează toate modurile posibile de reaşezare a persoanelor, astfel încât între oricare două persoane aflate în conflict să stea una sau cel mult două persoane.

Ex. Pentru n=4 programul trebuia să afişeze: 3 1 4 2

2 4 1 3

1. Se dă un alfabet care conţine v vocale şi c consoane. Se cere să se genereze toate cuvintele de lungime n care nu conţine trei vocale sau trei consoane alăturate.
2. Se consideră n piese de domino citite ca perechi de numere naturale, fiecare pe câte un rând de intrare. Să se afişeze toate soluţiile de aranjare a acestor piese într-un lanţ domino de lungime a, citită, fără a roti piesele. ( Un lanţ domino se alcătuieşte din piese domino astfel încât o piesă este urmată de alta a cărei primă jumătate coincide cu jumătatea a doua a piesei curente.
3. Se consideră n cuburi (max. 10), numerotate de la 1 la n, de laturi şi culori cunoscute. Să se numere şi şă se alcătuiască toate turnurile posibile din h cuburi astfel încât fiecare turn să aibă stabilitate (orice cub se aşează peste un altul cu latura mai mare sau egală cu a lui) şi să nu existe două cuburi consecutive de aceeaşi culoare. Soluţiile se vor afişa distinct, astfel ca utilizatorul să aibă timp să le citească, vor specifica numărul turnului şi vor cuprinde h grupuri de informaţii de tipul: *nr. cub, latură, culoare*.
4. Să se afişeze toate modurile de aranjare a elementelor unui şir dat de numere întregi, astfel încât în şirul rezultat să nu existe două elemente alăturate negative.
5. La un festival de muzică pop, s-au înscris n melodii, codificate 1, 2, 3,…, n (n>=4). Să se afişeze toate posibilităţile de a stabili ordinea intrării în concurs a melodiilor, ştiind că melodiile m1 şi m2 trebuie să evolueze a doua, respectiv penultima (m1, m2 {1, 2,…, n}.
6. La un concurs sportiv s-au înscris n concurenţi, numerotaţi 1, 2, …,n. Pentru fiecare concurent se cunoaşte ţara de origine (un şir de caractere). Să se afişeze toate posibilităţile de a stabili ordinea intrării în concurs a sportivilor, ştiind că:

* doi sportivi din aceeaşi ţară nu pot evolua unul după celălalt;
* trebuie respectată ordinea crescătoare a numerelor de concurs a sportivilor.

# II. Backtracking “cu sume”

1. Să se afişeze toate modurile de descompunere a unui număr natural n în sumă de p numere naturale distincte.
2. Scrieţi un program care să afişeze toate modalităţile prin care se poate plăti o sumă S folosind n bancnote de valori b1<b2<b3<…<bn. Se presupune că avem la dispoziţie oricâte bancnote din fiecare tip. Numerele n şi S, precum şi valorile bancnotelor se citesc de la tastatură, iar modalităţile de plată vor fi scrise în fişierul bani.out.
3. Să se generez toţi vectorii diferiţi, fiecare vector generat fiind format din n elemente cifre de 1 şi 0, suma elementelor sale neavând voie să depăşească valoarea S.
4. Să se afişeze toate şirurile de n paranteze care se închid corect (2<=n<=20)
5. Se consideră o mulţime M de numere naturale, având n elemente. Să se scrie un program care afişează toate submulţimile mulţimii M ale căror elemente însumate sunt egale cu o valoare naturală S dată.
6. Orice număr natural se poate scrie ca sumă de termeni numere naturale nenule. Astfel 4=3+1=2+2=2+1+1=1+1+1+1, deci în patru moduri. Să se scrie numărul n ca sumă de numere naturale şi să se afişeze numărul de posibilităţi descoperite.
7. Să se scrie în fişierul solutie.txt toate numerele (câte unul pe fiecare rând) de n cifre egale cu de k ori produsul cifrelor, n şi k fiind citite de la tastatură.
8. Se citeşte de la tastatură un număr natural nenul n<=8. Să se genereze toate şirurile de lungime n, formate din cifre de 0 şi 1, suma elementelor lor să fie egală cu o valoare dată S.
9. Există n şomeri care trebuie să se reprofileze, fiecare în câte una din n meserii. Şcolarizarea şomerului **i** în meseria **j** prezintă costul C[i,j], natural. Toate cele n meserii trebuie să fie ocupate de către cei n şomeri. Dându-se n şi matricea costurilor C(nxn) să se precizeze toate posibilităţile de atibuire a celor n meserii celor n şomeri, precum şi costul total al şcolarizării penru fiecare repartizare.
10. Să se afişeze toate numerele formate din cifre distincte cu proprietatea că suma cifrelor este S, dată. Soluţiile vor fi afişate într-un fişier de ieşire.

Ex : pentru S=3 se afişează : 102, 12, 120, 201, 21, 210, 3, 30.

1. Se dă o matrice pătratică A(nxn). Să se precizeze care este suma maximă care se poate obţine prin însumarea elementelor matricii aflate pe linii şi coloane diferite.
2. Pentru realizarea orarului unei clase de elevi trebuie să fie programate în fiecare zi, fie o oră de informatică din cele trei ore pe săptămână, fie o oră de fizică din cele două ore pe săptămână. În câte moduri se poate face această programare, ştiind că săptămâna are 5 zile lucrătoare?

O soluţie va fi o succesiune de 5 litere din mulţimea {I, F}, soluţia IFIIF semnificând faptul că luni este programată o oră de informatică, marţi una de fizică, miercuri şi joi câte una de informatică, vineri ultima de fizică.

1. La o admitere se dau n examene. La primul examen se pot obţine de la 0 la p1 puncte, la al doilea se pot obţine de la 1 la p2 puncte ş.a.m.d.. Pentru reuşită, un candidat trebuie să obţină cel puţin m puncte în total. Din fişierul f.in se citesc n, m şi şirul p1, p2, …,pn, naturale. Să se scrie în fişierul f.out toate variantele de de punctaje ce trebuie obţinute la cele n examene şi care conduc la reuşita la admitere.
2. Să se afişeze toate numerele de câte n cifre (0<n<=10) cu suma cifrelor egală cu S.
3. Într-o mare închisă sunt n porturi. Staţionarea într-un port se taxează printr-un cost asociat. Să se stabilească toate voiajele distincte prin p porturi (p<=n) care nu depăşesc un cost total de staţionare dat. Porturile se introduc prin denumiri, după ce s-au citit valorile pentru n şi p naturale, costurile şi costul total fiind date ca numere reale Soluţiile vor fi afişate secvenţial, pentru a putea fi citite (apăsare tastă), fiecare afişare conţinând: *nr. voiaj, lista denumirilor porturilor, costul total al voiajului.*.
4. Să se elaboreze toate variantele de chestionare ce se pot obţine dintr-un set total de n întrebări (n<=20) care să aibă între a şi b întrebări, iar întrebările să totalizeze minim p puncte şi maxim q puncte. Se citesc în ordine: numărul de întrebări, apoi fiecare întrebare, text de maximum 100 de caractere, urmată de punctajul său şi, la sfârşit, limitele a, b, p şi q.
5. Fiind dat un număr natural pozitiv n, să se producă la ieşire toate descompunerile sale ca sumă de numere prime.
6. In fişierul text numere.in se dau :

* pe prima linie un număr natural n<=15;
* pe a doua linie n numere naturale x[1], x[2], …,x[n] separate printr-un spaţiu ;
* pe a treia linia un număr natural S cu cel mult 9 cifre.

Se cere să se determine şirul de semne + şi – ce trebuiesc puse în în faţa numerelor x[1], x[2], …, x[n] pentru ca rezultatul expresiei să fie egal cu S. Aceste şiruri se vor scrie câte unul pe fiecare rând într-un fişier text solutie.out.

# III. Backtracking “colorat”

1. Se cere să se coloreze o hartă cu n ţări, folosindu-se numai 4 culori, oricare două ţări vecine să fie colorate diferit.
2. Se citesc 9 denumiri de culori (maximum 16 caractere fiecare) din care trebuie să se alcătuiască toate drapelele tricolore posibile, cu condiţia ca la mijloc să fie numai o culoare din ultimele două citite. Fiecare soluţie va apărea pe rând, câte una, la cererea utilizatorului (acţionarea unei taste).
3. Pe o tijă verticală se găsesc amestecate, m bile de n culori diferite. Să se separe aceste bile, punându-le pe n tije, conform celor n culori.
4. Avem la dispoziţie 6 culori, alb, galben, roşu, verde, albastru, negru. Să se precizeze toate drapelele tricolore care se pot proiecta, ştiind că orice drapel are prima culoare galben sau verde, cele trei culori de pe drapel sunt distincte, iar culoarea din mijloc trebuie să fie obligatoriu rosie. Afişarea se va face colorat, fiecare nume de culoare fiind scris explicit, în culoarea sa(ex: cuvântul roşu va fi scris cu culoarea roşie).
5. Utilizând m culori (maximum 5), numerotate de la 1 la m, să se coloreze în toate modurile posibile n piloni (maximum 10), care iniţiali sunt albi, aflaţi pe o suprafaţă oarecare, dacă aceştia se “văd” între ei şi doi piloni care se “văd” între ei nu trebuie să aibă aceeaşi culoare. Pilonii sunt în relaţie de se ”văd” între ei dacă orice linie imaginară care uneşte vârfurile a doi piloni nu trece printr-un câmp ocupat de alt pilon. Datele de intrare se compun din:

* un număr natural n reprezentând numărul de piloni
* perechi x, y , pe câte un rând fiecare, până la întâlnirea sfârşit de fişier(^Z), pereche ce semnifică faptul că x şi y se ”văd”
* un număr natural m, numărul de culori;
* denumirile culorilor, string-uri, câte unul pe fiecare rând.

1. Se consideră n mărgele de culori nu neapărat distincte. Să se scrie un program pentru afişarea tuturor posibilităţilor în care mărgelele se pot înşira astfel încât să nu existe două mărgele alăturate de aceeaşi culoare.

# IV. Alte probleme de backtracking

1. In vederea desfăşurării unui concurs, trebuie formată o delegaţie din p dintre cei n profesori ai unui liceu (p, n). Scrieţi un program care pentru valorile citite de la tastatură ale lui n şi p, va genera, folosind backtracking, în fişierul comisie.bac toate posibilitătile de alcătuire a delegaţiei. Fiecare soluţie va fi afişată pe o linie separată în fişierul de ieşire, elementele fiind separate prin spaţii. Consideraţi profesorii numerotaţi de la 1 la n. Programul va fi comentat.
2. Generaţi prin metoda backtracking toate numerele naturale cu n cifre (1<=n<=9) care sunt “supermultiple” de p (p>1). Un număr este “supermultiplu” de p, dacă atât numărul cât şi toate numerele obţinute din el prin tăierea succesivă a cifrelor sale începând cu cifra unităţilor sunt multiple

de p. Valorile n şi p se introduc de la tastatură şi sunt corecte. Descrieţi algoritmul pe care îl folosiţi.

1. Numim numărul x « superprim « dacă atât x cât şi toate numerele obţinute prin amestecarea cifrelor lui x sunt numere prime. Pentru o valoare n citită, scrieţi un program care să determine dacă acesta reprezintă un număr superprim sau nu.
2. Ex: pentru n=3 şi p=2 numărul 246 este “supermultiplu” de 2 deoarece numerele 246, 24, 2 sunt multiple de p.
3. Se citeşte un număr natural n. Să se genereze toate numerele care au în reprezentarea lor binară atâtea cifre de 1 şi de 0 câte are şi reprezentare numărului n în baza 2.
4. Se citesc de la tastaură două numere naturale n şi m (o<m<12). Să se afişeze toate şirurile de n litere distincte, litere alese dintre primele m ale alfabetului englez.

Ex : pentru n=2 şi m=4 se afişează, nu neapărat în această ordine, şirurile :AB, BA, AC, CA, AD, DA, BC, CB, BD, DB, CD.

1. Câţiva copii trebuie să fie vizitaţi de Moş Crăciun. Scrieţi un program care determină toate modurile diferite în care pot fi ei aşzaţi în lista lui Moş Crăciun, astfel încât să fie toţi vizitaţi în ordinea descrescătoare a vârstei lor.

Se citesc de la tastatură : n, numărul de copii (0<n<10), apoi numele şi vârsta fiecăruia dintre cei n copii. Se scriu în fişierul craciun.txt, pe linii diferite, liste cu numele copiilor, în irdinea în care vor fi vizitaţi de Moş Crăciun. O listă este formată din toate cele n nume ale copiilor, într-o anumită ordine, orice două nume succesive fiind despărţite printr-un spaţiu.

Ex : pentru datele de intrare

n=4

Dan 6

Cristina 4

Corina 2

Iulia 4

se scriu în fişierul craciun.txt următoarele soluţii:

Corina Iulia Cristina Dan

Corina Cristina Iulia Dan

1. Să se scrie în fişierul soluţii.dat toate numerele de n (0<n<10) cifre, care adunate fiecare cu numărul obţinut prin inversarea ordinii cifrelor sale dau un număr pătrat perfect.
2. Se dă un număr natural n. Se cere să se genereze toate matricele nxn având ca elemente numere distincte din 1..nxn astfel încât nici un element să nu aibă aceeaşi paritate cu vecinii săi (vecinii unui element se considera pe directiile N,S,E,V).
3. Un patron vrea sa-si aprovizioneze depozitul en-gross cu anumite produse. El are de ales dintre n centre de aprovizionare.Fiecare centru dispune de o cantitate p[i] de produse. Stiind că:

- viziteaza centrele în ordinea 1🡪n (crescătoare)

- de la un centru ia toata marfa

- de la un centru trebuie sa ia o cantitate mai mare decat de la centrul anterior

ajutaţi-l pe patron să găsească cel mai lung drum ce satisface conditiile de mai sus.

1. Se da o lista de n cuvinte. Sa se formeze cel mai lung sir in care fiecare cuvant incepe cu litera cu care se termina predecesorul.
2. Sa se genereze toate permutarile de dimensiune n cu valori in 1..n cu proprietatea ca oricare ar fi 2<=i<=n exista un 1<=j<=i astfel incat |v(i)-v(j)|=1.
3. O persoană are la dispoziţie un rucsac cu o capacitate de G unitaţi de greutate şi intenţionează să efectueze un transport in urma căruia să obţină un câştig. Persoanei i se pun la dispoziţie N obiecte.Pentru fiecare obiect se cunoaşte greutatea(mai mică decât capacitatea rucsacului) şi câştigul obţinut in urma transportului său. Să se afişeze toate modalităţile de a-şi încărca rucsacul cu obiecte dintre cele date, astfel încât să obţină un câştig c, dat, ştiind că obiectele nu se pot tăia.
4. Sa se dispună pe cele 12 muchii ale unui cub toate numerele de la 1 la 12, astfel incât suma numerelor aflate pe muchiile unei fete să fie aceeaşi pentru toate feţele.
5. Se dă un număr natural par N .Să se afeşeze toate şirurile de N paranteze care se inchid corect.
6. Se consideră o mulţime de N elemente şi un număr natural k nenul.Să se calculeze câte submulţimi cu k elemente satisfac pe rând condiţiile de mai jos şi să se afişeze aceste submulţimi :

a) conţin p obiecte date

b)nu conţin nici unul din q obiecte date

c)conţin exact un obiect dat,dar nu conţin un altul

d)conţin exact un obiect din p obiecte date

e)conţin cel puţin un obiect din p obiecte date

f)conţin r obiecte din p obiecte date,dar nu conţin alte q obiecte date

17. Se dau N puncte albe şi N puncte negre in plan ,de coordonate intregi .Fiecare punct alb se uneşte cu

câte un punct negru ,astfel incât din fiecare punct,fie el alb sau negru, pleacă exact un segment.Să se determine o astfel de configuraţie de segmente incât oricare două segmente să nu se intersecteze.Se citesc 2N perechi de coordonate corespunzând punctelor.

18. Se consideră n puncte in plan , de coordonate reale.Elaboraţi un program care selectează din aceste puncte vârfurile unui pătrat,ce conţine numărul maximal de puncte din cele date.Afişaţi coordonatele punctelor selectate şi numărul de puncte incluse in pătratul respectiv.

19. Fiind dat un număr natural N şi un vector cu N componente intregi, se cer următoarele :

-să se determine toate subşirurile crescătoare de lungime[n/5]

-să se calculeze p(1)+p(2)+….+p(k),unde p(k) reprezintă numărul subşirurilor crescătoare de lungime k

20. Se citeşte de la tastatură un cuvânt c, format din n litere mari (n<=15). Presupunând că sunt generate în ordine lexicografică (ordinea « de dicţionar ») toate cuvintele distincte ce se pot forma cu exact n litere din mulţimea {A, D, M, O}, să se scrie un program care verifică dacă c se află sau nu printre cuvintele generate şi, în caz afirmativ, afişează al câtelea este în listă.

Se va afisa pe ecran NU dacă c nu se află printre cuvintele generate> Dacă c se află printre cuvintele generate se va afisa pe ecran mesajul DA, urmat, pe raândul următor, de numărul cerut.

Exemplu : penrt c=ADA se vor afişa pe ecran liniile :

DA

5

Se observă că şirul citit are 3 litere. Succesiunea de cuvinte formate cu 3 litere din mulţimea {A, D, M, O}, cuvinte aflate în ordine lexicografica, este : AAA, AAD, AAM, AAO, ADA, ADD, ADM, ADO, AMA, AMD, AMM, AMO, AOA, AOD, AOM, AOO, DAA, DAD, DAM, etc. Deci ADA se află în şirul de cuvinte şi este al cincilea.

21. Se citeşte de la tastatură un număr natural (n<15). Să se genereze toate şirurile formate din numerele 1, 2, 3, …, n astfel încât niciuna dintre valori să nu lipsească şi să nu apară de două ori într-un şir, iar valorile impare să se găsească întotdeauna « la locul lor »: 1 pe prima pozitie din şir, 3 pe a treia poziţie din şir, 5 pe a cincea poziţie din sir, etc. Fiecare şir se va afisa pe un rând al ecranului, cu spaţii între elementele ce-l formează. De exemplu, penrtu n=6, se vor afişa pe ecran ( nu neapărat în această ordine) şirurile :

1 2 3 4 5 6

1 2 3 6 5 4

1 4 3 2 5 6

1 4 3 6 5 2

1 6 3 2 5 4

1 6 3 4 5 2

Se observă că în fiecare şir locul valorilor impare este neschimbat.

22. Se citeşte de la tastatură un număr natural (n<15). Să se genereze toate şirurile formate din numerele 1, 2, 3, …, n astfel încât niciuna dintre valori să nu lipsească şi să nu apară de două ori într-un şir, iar valorile pare să se găsească întotdeauna « la locul lor » : 2 pe a doua pozitie din şir, 4 pe a patra poziţie din şir, 6 pe a şasea poziţie din sir, etc. Fiecare şir se va afisa pe un rând al ecranului, cu spaţii între elementele ce-l formează. De exemplu, penrtu n=6, se vor afişa pe ecran ( nu neapărat în această ordine) şirurile :

1 2 3 4 5 6

1 2 5 4 3 6

3 2 1 4 5 6

3 2 5 4 2 6

5 2 1 4 3 6

5 2 3 4 1 6

Se observă că în fiecare şir locul valorilor pare este neschimbat.

23. Se citesc de la tastatură un număr natural n (n<20) şi apoi un cuvânt cuv format din cel mult 10 litere mici ale alfabetului englez. Să se scrie un program care afişează în fişierul CUVINTE.TXT n cuvinte distincte care se pot forma prin amestecarea literelor cuvântului dat. Cuvântul nu trebuie să se afle printre cele n cuvinte generate. Fiecare cuvânt obţinut va fi afisat pe o linie separată din fiier. Dacă nu există posibilitatea formării a n cuvinte distincte cu proprietatea cerută, se va scrie în fişier mesajul EROARE.

De exemplu : pentru n=4 şi cuv=examen, 4 soluţii posibile care se pot scrie în fîsierul CUVINTE.TXT sunt :

amenex

eaexmn

emanex

xemena

Se observă că sunt şi alte posibilităţî de a alege 4 cuvinte care respectă condiţiile din enunţ, penrtu a fi scrise în fişier.

24. Se citesc de la tastatură un număr natural s (0<s<5000) reprezentând o sumă de bani exprimată în mii de lei. Să se determine un mod de plată a sumei s ştiind c avem la dispoziţie 9 monede de o mie, 8 monede de 5 mii, 5 bancnote de 10 mii, 8 bancnote de 50 de mii şi 45 bancnote de 100 de mii de lei. Se va afişa pe ecran modalitatea de plată în formatul sugerat prin exemplul următor sau mesajul IMPOSIBIL dacă nu este posibilă palta sumei cu monedele disponibile.

Pentru s=3074 (3074 mii de lei), se poate afişa soluţia:

3074=3x100+1x50+2x10+4x1.

Se observă că sunt mai multe ţi de plată, însă problema solicită determinarea doar a uneia dintre posibilităţi.